Filing Date: November 26, 2003
Assignee: Intel Corporation

P16657

APPLICATION FOR UNITED STATES LETTERS PATENT

INVENTORS: Shih-Lien L. LU, Dinesh SOMASEKHAR and Yibin YE

TITLE: SYSTOLIC MEMORY ARRAYS

ATTORNEYS & ADDRESS: FLESHNER & KIM, LLP
P. O. Box 221200
Chantilly, VA 20153-1200

DOCKET NO.: INTEL-0034

Systolic Memory Arrays

BACKGROUND OF THE INVENTION

1. Field of the Invention

[1] The present invention is directed towards a systolic memory array (SMA) and more particularly to a SMA that enables the access of memory arrays that are subdivided into a plurality of banks and these banks may be accessed in a pipelined manner.

2. Background of the Related Art

[2] One perennial goal among circuit and system designers is to improve memory bandwidth and access times. One way of achieving this goal, while simultaneously improving bandwidths and access times in memory devices, is to divide or compartmentalize an internal memory structure into a plurality of blocks that expand or increase the width of the data bus that accesses the memory structure.

[3] Memory structures used in microprocessors and computing systems (e.g. a processor and memory) are growing rapidly in size and capability to accommodate the larger, proliferating new applications of increasing complexity and to improve processor performance. In general, systolic structures are used for mapping computations or processes into hardware structures and SMAs are used to map computations or processes into memory structures.

[4] There is typically a direct relationship between a memory's capacity and its physical size, where a larger memory results in a larger physical size and a smaller memory

results in a smaller physical size. This larger physical size increases the access time due to the inherent wiring delay present in longer wires and communication paths associated with the larger size. This makes accessing data and information stored in a memory structure within a short time or an otherwise acceptable time an increasingly difficult process.

[5] Therefore, the various exemplary embodiments of the present invention address the disadvantages mentioned above and disclose a memory array that includes a plurality of multiple banks, which are adjustable in size (e.g. they can be made smaller or larger). These banks have shared address lines and are accessed in a pipelined fashion. After a certain latency or delay transpires, data stored in the banks at every clock cycle can be read out. Memory accesses to this memory array are sustainable for both reads and writes.

[6] The various exemplary embodiments of the present invention permits memory arrays subdivided in banks to be accessed in a pipelined fashion. This approach achieves a much higher sustainable memory bandwidth and possibly a shorter average access time than what the individual banks' provides if they were accessed with shared non-pipelined buses. This design also alleviates the problem of driving long global bit lines in larger memories. Read access of this type of pipelined memory will exhibit physical locality properties and have variable latency. Banks that are located closer to an access port will have shorter access time than banks that are located farther away. Additionally, systolic memories are easier to implement because of their modular designs and they are also more cost effective to produce because of this modular characteristics.

BRIEF DESCRIPTION OF THE DRAWINGS

[7] The invention will be described in detail with reference to the following drawings in which like reference numerals refer to like elements wherein:

[8] Figure 1(a) is an exemplary embodiment of a pipelined memory array;

[9] Figure 1(b) is an exemplary embodiment of a systolic memory array;

[10] Figure 2(a) is an exemplary illustration of an Read Address/Data Movement;

[11] Figure 2(b) is an exemplary illustration of a Write Address/Data Movement

[12] Figure 3 is an exemplary timing diagram of read pipeline timing;

[13] Figure 4 is an exemplary timing diagram of write pipeline timing;

[14] Figure 5 is an exemplary timing diagram of a read after write operation;

[15] Figure 6 is an exemplary timing diagram of a write after read operation; and

[16] Figure 7 is an exemplary diagram of an exemplary computer system implementing a systolic memory array.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[17] Additional advantages, and features of the invention will be set forth in part in the description which follows and in part will become apparent to those having ordinary skill in the art upon examination of the following or may be learned from practice of the invention. The advantages of the invention may be realized and attained as particularly pointed out in the appended claims.

[18] A pipelined memory is a memory that is arranged in a pipelined manner. Those skilled in the art will appreciate that the various embodiments of the present

invention may be implemented in different types of memory and storage media (e.g. DRAM, SRAM, magnetic, optical etc.).

[19] The various exemplary embodiments of the present invention include a plurality of memory arrays that have a memory architecture similar to an apparatus architecture that is coupled to the plurality of memory arrays. For example, if the apparatus architecture is a pipelined microprocessor architecture, then the memory architecture that is coupled to the pipelined microprocessor architecture would utilize a pipelined architecture. Similarly, if the apparatus architecture is a non-pipelined microprocessor architecture, then the memory architecture that is coupled to the non-pipelined microprocessor architecture is also a non-pipelined architecture. In other words, the memory architecture is structured to be similar to the apparatus or device that is coupled to the memory.

[20] A generic memory is organized and arranged into a two dimensional (2-D) array including rows and columns that are multiplexed together. In one exemplary embodiment, the array arrangement has word lines running horizontally and bit lines running vertically and the arrays of memory blocks are accessed independently. The memory is accessed by assigning a word address. Then the word address is decoded to activate a corresponding one of the word lines by using the bit lines and data is retrieved.

[21] Data stored on a horizontal line is read out in parallel and the data may be reduced in width through a selection or grouping operation. In one exemplary embodiment, the memory array is sub-divided into banks horizontally. Two types of pipeline may be employed for performing this access.

[22] First, there is the address pipe that is used to pump the address into each bank in the memory array. Second, there is also another data pipe that allows the data to go in and out of the array. Figure 1(a) illustrates one exemplary embodiment of this type of pipelined memory array.

[23] In Figure 1(a), Bank0 (102), Bank1(104), ... Bankn-2 (106), and Bankn-1 (108) are coupled together in a pipelined memory array 100 so that the memory bank locations are addressable and data can be read into and out of the memory without any data collisions occurring. Data is accessed in a predetermined order to dynamically prevent collisions from occurring. Timing diagrams are developed to reflect and characterize memory operations and one specific application of these timing diagrams is to help avoid these data collisions. The timing diagram also helps to characterize the flow of data into and out of the memory. Once the data flow is understood, data access can be predetermined in accordance with this data flow to avoid collisions and allow the memory to function in a more optimized manner.

[24] During operation, the pipelined memory operates like a pipelined processor and some of the same operating principles that have been applied to pipelined processors (e.g. super-scalar processing and out of order processing) are applied to memory structures and their operation in accordance with the various embodiments of the present invention. The pipelined memory will support out-of-order data read, write and access operations.

[25] Those skilled in the art will realize that there is no need to restrict the pipeline to one-dimension only. Multiple dimension pipelines can be employed to overcome the long

wiring delay problem and still remain within the spirit and scope of the claimed embodiments of the present invention.

[26] Figure 1(b) illustrates an exemplary embodiment of how multiple arrays can also be pipelined vertically. In Figure 1(b), two systolic memory arrays, Array0 (110) and Arrayn-1 (112) are shown. In Array0 (110), Bank0 (114), Bank1(116), ... Bankn-2 (118), and Bankn-1 (120) are coupled together in a pipelined memory array so that the memory bank locations are addressable and data can be read into and out of memory. A pipeline register 111 interfaces with Array0 and pipeline register 113 interfaces with Arrayn-1.

[27] In Arrayn-1 (112), Bank0 (122), Bank1(124), ... Bankn-2 (126), and Bankn-1 (128) are coupled together in a pipelined memory array so that the memory bank locations are addressable and data can be read into and out of memory. A pipeline register 113 is used to interface with Array0.

[28] One advantage of the exemplary design illustrated is that all peripheral access is from one side (in this case the left side as shown in Figure 1(b)). In this exemplary embodiment, at least two data pipes are needed - one data pipe is used for reading and one data pipe is used for writing. This is done to reduce the read-write and write-read turn-around time and avoid the contention of resources. However, those skilled in the art will realize that additional numbers of data pipes for reading and writing operations and interfacing with the memory arrays may also be used, without departing from the spirit and scope of the present invention.

[29] Each bank will have a mechanism for supporting addressing and data operations (e.g. pipeline registers for supporting addressing and data operations). The

number of pipeline stages selected depends on the access latency of each bank and the desired throughput rate.

[30] For a given throughput requirement, a clock frequency and the data pipe width for the pipeline are determined. The individual bank access latency is converted into a number of pipeline clock cycles. In one exemplary embodiment, there should be the same number of pipeline stages as the number of clock cycles.

[31] For example, if the desired throughput is 8 GB per second and the datapath width is 8B, then the clock frequency is 1GHz. If the access latency is 8 nano-seconds, then there will be 8 pipeline stages. Writing is done by pumping the address together with the data to be written.

[32] Data for the right most bank enters the pipeline first, while data for the left most bank enters last. However, the bank to the left is written first while the bank on the right is written last. Reading is done by pumping the address once and allowing the address to flow through the address pipe to reach individual banks one cycle at a time. Whenever a bank receives the read address, access to the bank is started. Therefore, the access latency for an ith bank is represented by 2i+L. It will take i cycles to allow the address to reach the desired ith bank. It will take L cycles of latency to access the memory. It will take i cycles again, to allow the data to come out from the ith bank through the read data pipeline. When data is ready at the bank memory, it needs to enter the read data pipeline.

[33] In order to avoid data collision during memory operation, the second read access of a plurality of consecutive reads must delay the placement of read result on the read data pipeline by one cycle. This delay is represented in the timing diagrams by the idle time

that is inserted into the memory operation. This will ensure that no data collisions will occur. Notice that results from different banks having different addresses will be interleaved.

[34] Figure 2 depicts an exemplary illustration of the read and write processes. Figure 2(a) shows the read operations. Each address of the bank arrives at the designated bank and the data is read out from there and placed on the data bus after the array access time. The thick arrow shown in the figure illustrates an exemplary address/data path.

[35] Figure 2(b) illustrates the write operation. Addresses continue to enter into the array and a corresponding data travel on the data bus in sync with the address. A control signal (not shown) enables the write process to occur at the appropriate time.

[36] Figures 3 - 6 that follow illustrate various timing diagrams of an exemplary pipelined memory for different types of memory accesses.

[37] The various exemplary embodiments of the present invention that will be discussed have 8 banks in each array. Those skilled in the art will appreciate that more than 8 banks may be used without departing from the spirit and scope of the present invention. In these Figures, the horizontal axis is time. Data at the ith bank is denoted as Di. Addresses entering the pipeline are labeled as Ai. For example Al is the first access address that a user would like to address, while A2 is the second access address to be addressed and so on. In this example, data coming out from odd addresses (Al, A3 ...) will interleave with data from even addresses (A2, A4 ...) after initial filling of the pipe.

[38] The data and addresses are in the following order — D1(Al), D2(Al), D3(Al), D4(A1), D5(Al), D1(A2), D6(Al), D2(A2), D7(Al), D3(A2), D8(Al), D4(A2), D1(A3), D,

D5(A2) .... In this exemplary embodiment, every 8 cycles, a new address can enter this pipelined array memory. Data can be read out and written at every cycle after an initial latency.

[39] In these exemplary timing diagrams, L refers to a latency associated with each of the memory banks. The specific latency value that is ultimately selected should be the same for each bank. However, those skilled in the art will appreciate that different latencies may be selected for different memory arrangements, schemes or layouts, without departing from the spirit and scope of the present invention.

[40] Figure 7 is an exemplary diagram of an exemplary computer system including storage media using systolic memory arrays in accordance with the exemplary embodiments of the present invention. The computer system may include a microprocessor 2, which includes many sub-blocks, such as an arithmetic logic unit (ALU) (4) and an on-die cache 6. The microprocessor 2 may also communicate to other levels of cache, such as off-die cache 8. Higher memory hierarchy levels such as system memory 10 (e.g. RAM), are accessed via a host bus 12 and chipset 14. In addition, other off-die functional units, such as a graphical interface 16 and network interface 18, to name just a few, may communicate and interoperate with the microprocessor 2.

[41] A systolic memory array in accordance with the various exemplary embodiments of the present invention may be used in the on-die cache 6, the off-die cache 8 and the RAM 10 or in any other location that memory or storage media is used in the computer system.

[42] The foregoing embodiments and advantages are merely exemplary and are not to be construed as limiting the present invention. The present teaching can be readily applied to other types of apparatuses. The description of the present invention is intended to be illustrative, and not to limit the scope of the claims. Many alternatives, modifications, and variations will be apparent to those skilled in the art. In the claims, means-plus-function clauses are intended to cover the structures described herein as performing the recited function and not only structural equivalents but also equivalent structures.